FYP11001 – Interim Report

Mobile Assistant for Accessibility

Enhancing the Communication Ability of the Hearing Impaired

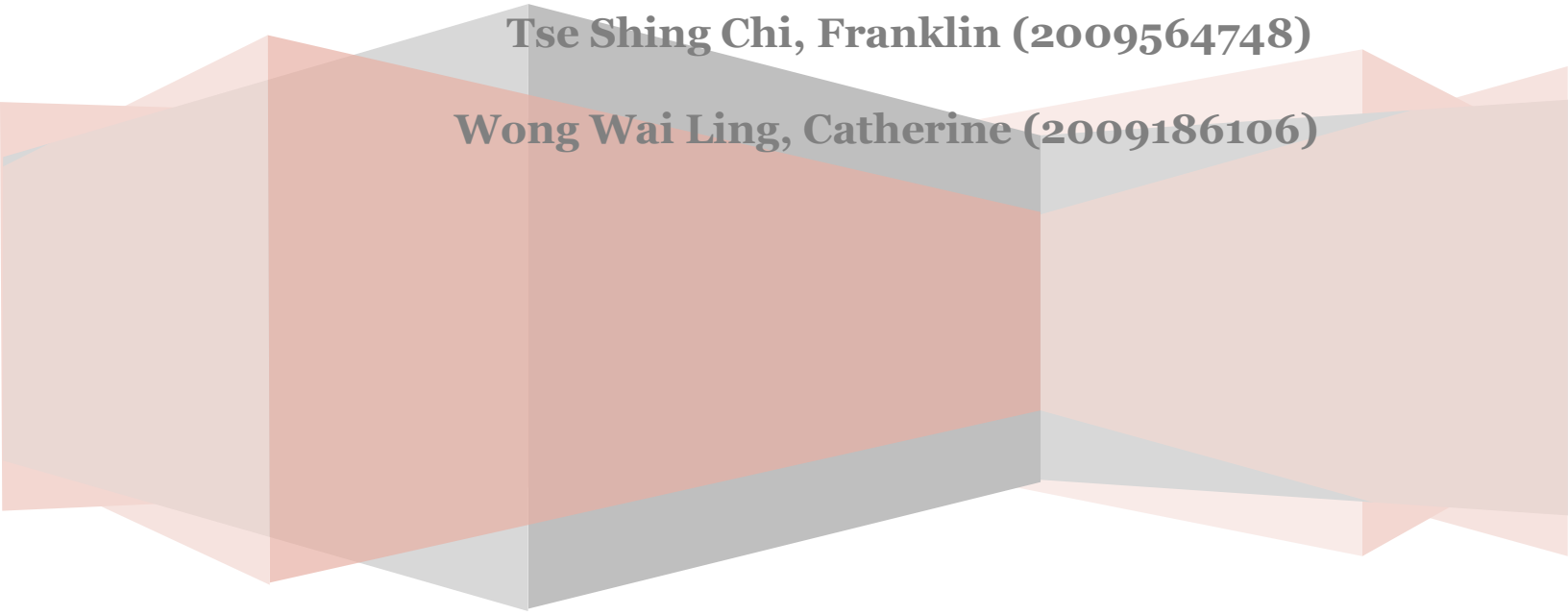
Supervisor: Dr. Vincent Lau

Team Members: Lau Kam Lun, Eric (2009572812)

Law Chi Lok, Cyrus (2009559420)

Tse Shing Chi, Franklin (2009564748)

Wong Wai Ling, Catherine (2009186106)



Revision

Version	Description	Date
0.1	Draft	23 Dec 2011
1.0	First Public Release	09 Jan 2012

Table of Contents

1 Introduction 6

1.1 Background 6

1.2 Project Overview 7

2 Feasibility Study 8

2.1 Phone Call Recording 8

2.1.1 Result 8

2.1.2 Evaluation 8

2.1.3 Resolution 8

2.2 Streaming Recorded Audio 8

2.2.1 Result 8

2.2.2 Evaluation 9

2.2.3 Resolution 9

2.3 Speech Recognition (Speech-to-Text) 9

2.3.1 Result 9

2.3.2 Evaluation 10

2.3.3 Resolution 10

2.4 Text-to-Speech 10

2.4.1 Result 10

2.4.2 Evaluation 10

2.4.3 Resolution 11

2.5 Splitting up Recorded Audio Stream 11

2.5.1 Result 11

2.5.2 Evaluation 11

2.5.3 Resolution 11

2.6 Multi-Language Support 11
2.6.1 Result 11
2.6.2 Evaluation 12
2.6.3 Resolution 12
2.7 Gender 12
2.7.1 Result 12
2.7.2 Evaluation 12
2.7.3 Resolution 12
3 Development Environment 13
3.1 Platform 13
3.2 Development Tools 13
3.2.1 Client-side Development 13
3.2.2 Server-side Development 14
4 Project Implementation 15
4.1 InCom 15
4.1.1 System Architecture 15
4.1.2 Text-to-Speech 16
4.1.3 User Interface 16
4.1.4 Dual-Tone Multi-Frequency Signaling (DTMF) Input 17
4.1.5 Speech-to-Text 17
4.1.6 Network Data Usage 19
4.1.7 Known Issue 19
4.2 SS 20
4.2.1 Sentence Structure Research 20
4.2.2 Emergency Service 26

5 Dependencies 38
6 Software Documentation 38
6.1 Application Manual 38
6.2 Javadoc 38
7 Quality Assurance 39
7.1 Unit Testing 39
7.2 Revision Control 39
7.3 Usability Testing 39
8 Financial Report 40
9 Schedule 41
10 Work Division 42
11 Coming Development 43
12 Conclusion 44
13 References 45
14 Appendix I: Questionnaire 46

1 Introduction

1.1 Background

The hearing impaired are people who have difficulties in hearing or are not able to hear any sound. Their communication therefore relies on sign language instead of talking using voice. Since general public normally do not know sign language, the hearing impaired faces challenges in communicating with and, when necessary, seeking helps from general public.

Many telephone hotlines use an interactive voice response system to process operations or requests such as appointment booking and customer services. The hearing impaired are unable to use IVR system because of their limitation. Service providers usually solve it by developing an online system or a SMS service specifically for the hearing impaired. This approach is costly and can be inconvenient.

On the other hand, the language proficiency of the hearing impaired is usually worse than general public, as oral communication is an important part of a language and they could not learn through listening. In addition, the sign language they use has a different structure, so it is difficult for them to learn the language used by general public.

Hearing impaired in Hong Kong mainly use SMS, video call and internet video conferencing software to communicate. However, video conferencing is costly for mobile phones, and their SMS are usually not easy to understand as the grammar of the message is not correct.

1.2 Project Overview

Our project aims to tackle the inconvenience of the hearing impaired in using phone call and seeking help from general public. Those inconveniences are usually brought by their inability on hearing voice, so our project is going to develop a mobile application to act as their ear, helping them to listen to phone and people conversation.

Our mobile application is called InCom and there are two components:

The first component is an application called InCom that converts speech from the phone or microphone into words and displays the words on the screen. If users want to reply, they can type word inside the application and the application will change those words to speech, or use the keypad for responding to telephone hotline IVR systems.

The second component is an input method called SS (Smart Sentence) installed to the mobile phone, which provides many packaged sentence structures for the hearing impaired to type sentence with a correct grammar and much faster.

We believe the communication issues should not limit the connection between the hearing impaired and the others. The distance between the hearing impaired and the general public should be shortened and the hearing impaired could gain more benefit in their daily life.

2 Feasibility Study

As this is our first time to develop an Android application, and voice processing is new to us, we have conducted a study in October 2011 to evaluate the feasibility of certain essential features of our project and find out solutions or workarounds to any problem arisen.

2.1 Phone Call Recording

2.1.1 Result

Android SDK supports phone call recording, but some devices may not support that. Preliminary tests show that the phone we bought supports phone call recording, but it seems to record sound from the microphone at the same time.

2.1.2 Evaluation

- Our application may not work on some devices.
- Having sound from microphone recorded adds noise to the recording, which may reduce the accuracy of speech recognition.
- It is expected that most users contact an interactive voice response (IVR) system indoor, such as their home, and thus the background noise should be low.

2.1.3 Resolution

- No action required.

2.2 Streaming Recorded Audio

2.2.1 Result

The commonly-used `android.media.MediaRecorder` class does not support audio streaming—it always outputs the recording to a file. The more native

`android.media.AudioRecord` class supports streaming and thus should be used instead of `MediaRecorder`.

2.2.2 Evaluation

- Although `AudioRecord` is more difficult to use, no significant impact on the project is expected.

2.2.3 Resolution

- Use `AudioRecord`.

2.3 Speech Recognition (Speech-to-Text)

2.3.1 Result

The `android.speech` package does not provide a method for sending an audio stream to Google web service for speech recognition. It only provides a dialog asking the user to speak and then internally passes the recording to Google. As a result, a custom web service has to be built for speech recognition. Further study, however, finds that there is no free Cantonese speech recognition with dictation ability available. The following workarounds are identified:

- Use a Cantonese speech recognition engine that supports custom grammar, and then write grammars for each supported IVR systems.
- Withdraw Cantonese support and use an English or Putonghua speech recognition engine that supports dictation instead.

2.3.2 Evaluation

- Cantonese is the most widely-used language in Hong Kong. Even though the hearing impaired may not care the language as they cannot hear, most IVR systems support Cantonese and thus supporting Cantonese ensures maximum capability.
- Writing grammar improves the accuracy of speech recognition significantly. However, grammar is difficult to be generic and each IVR system needs its own grammar. Furthermore, any change to the IVR system may require an update to the grammar.
- If Cantonese cannot be supported, Putonghua is preferred because it is still Chinese, yet conversion from Simplified Chinese to Traditional Chinese may be needed.

2.3.3 Resolution

1. A set of Cantonese grammars for a selected IVR system will be built to evaluate the workload and manpower required.
2. Look for Putonghua speech recognition engine with dictation ability.

2.4 Text-to-Speech

2.4.1 Result

Most, if not all, Android phones do not support Cantonese text-to-speech by default. A custom web service has to be built for this purpose. The Microsoft Speech Platform for Servers, which is free and supports 26 languages including Cantonese and Putonghua, is a good choice.

2.4.2 Evaluation

- A server running Windows is required for hosting the web service. Currently, our budget allows us to apply for 1 server.

2.4.3 Resolution

- Apply for a server running Windows Server 2008 R2, develop a text-to-speech web service that supports English (United Kingdom), Cantonese (Hong Kong) and Putonghua, using Microsoft Speech Platform for Servers.

2.5 Splitting up Recorded Audio Stream

2.5.1 Result

As `android.media.AudioRecord` encodes the recording in linear pulse-code modulation (LPCM) format, which is widely-supported, splitting up the recorded audio stream for each silence point can be done on the phone or server-side.

Defining and identifying silence points requires knowledge on the LPCM format and volume analysis.

2.5.2 Evaluation

- Whether split-ups are done on the phone or server-side depends on performance. Since the server has higher computational power and more resources, it seems better to do the split-ups there.

2.5.3 Resolution

- Split-ups are done server-side.

2.6 Multi-Language Support

2.6.1 Result

Usually 1 speech-to-text engine supports only 1 language. To support multiple languages, each language would require its own engine. In addition, it has been difficult to find engines for non-English languages.

2.6.2 Evaluation

- The target users of the application mostly use Traditional Chinese. Not supporting other languages should not have significant impact.

2.6.3 Resolution

- This feature will not be implemented.

2.7 Gender

2.7.1 Result

Gender support relies on support from the text-to-speech engines. Current finding shows that most free engines only support 1 gender and do not allow users to choose which gender to use.

2.7.2 Evaluation

- Outputting the voice of a gender different from the user's may cause some confusion. However, the impact should not be major.

2.7.3 Resolution

- This feature will not be implemented.

3 Development Environment

3.1 Platform

After comparing with several mobile phone platforms, we decided to target our application on Android phone. It is because Android application can be written in Java, a language that we are familiar with, and Android is getting increasingly common in Hong Kong.

3.2 Development Tools

3.2.1 Client-side Development

3.2.1.1 Software Development Kit

There are 2 development kits for Android application development: Android SDK and Android NDK. The former one allows developers to develop their application in Java while the latter one allows developers to use C/C++. Both tools result in the developed application being run on the Android virtual machine. For simplicity and easier development, the Android SDK is selected.

3.2.1.2 Integrated Development Environment

InCom is developed using Eclipse, an open-source and multi-language software development environment, with a plugin called Android Development Tools (ADT) provided officially by Android project. ADT handles Android project creation and integrates Android SDK tools into Eclipse, which eases our workload during development.

3.2.2 Server-side Development

As the Microsoft Speech Platform for Servers is a .NET Framework library, the development tools involved are generally fixed: .NET Framework and Visual Studio. C# is the chosen language for developing the web services on server.

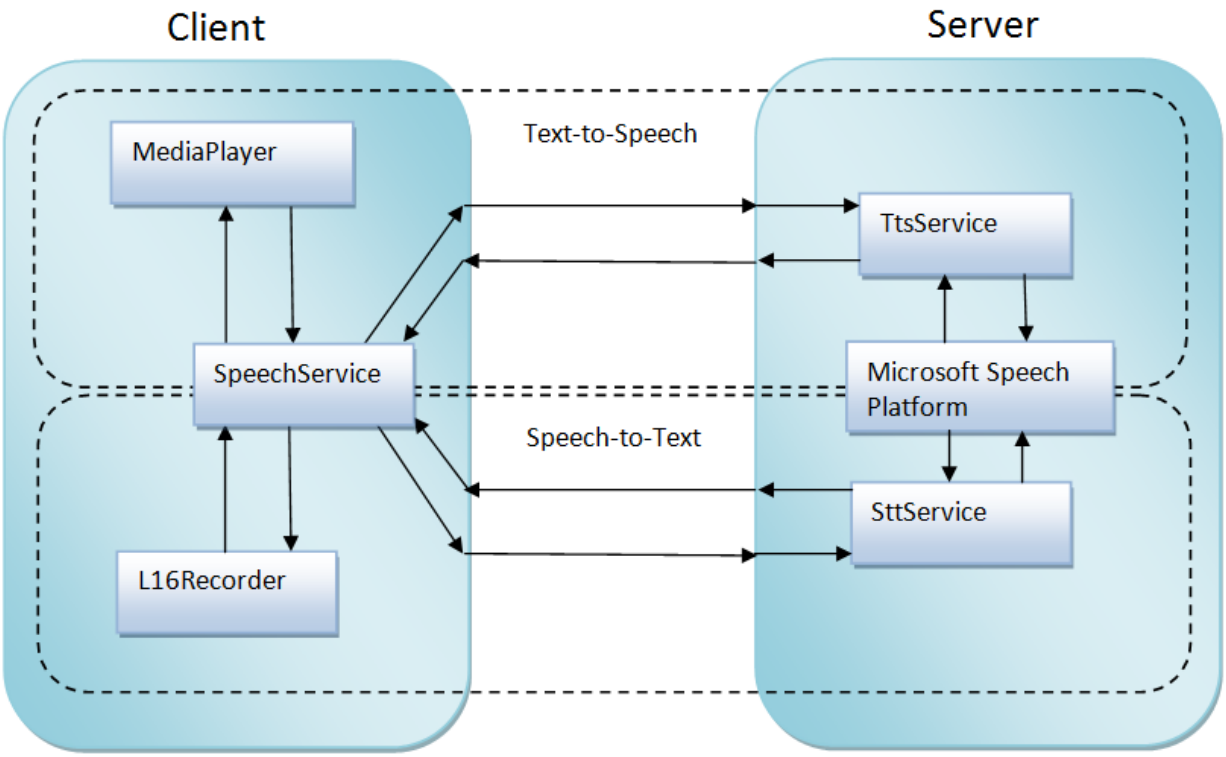
4 Project Implementation

The application that our team is developing is called InCom. This application consists of two components: a speech and text conversion application that is also called InCom, and an input method called SS.

4.1 InCom

InCom is a speech and text conversion application. It comes from the words “Interactive” and “Communication”. “In” can also mean “Intelligent”. The purpose of developing InCom is to allow the hearing impaired to communicate with others through voice. In this way, their communication skills can be enhanced.

4.1.1 System Architecture



4.1.2 Text-to-Speech

Text-to-Speech depends on the Microsoft Speech Platform for Servers, which is a .NET Framework library that runs on Microsoft Windows. To use the library, a web service was built using Windows Communication Foundation (WCF). The web service is based on HTTP, which is widely used and supported. It accepts a POST request with UTF-8 encoded text and responds with the speech of the text in WAVE format.

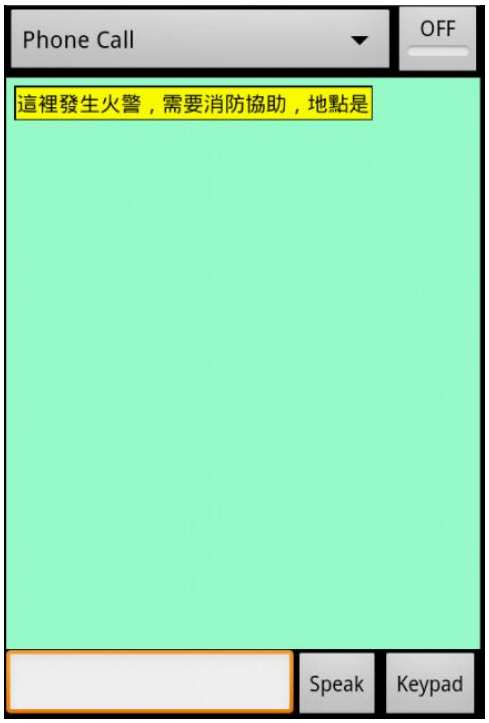
Currently, the text-to-speech service supports 3 languages, English (United Kingdom), Cantonese (Hong Kong) and Putonghua, as specified in the URL by user agent.

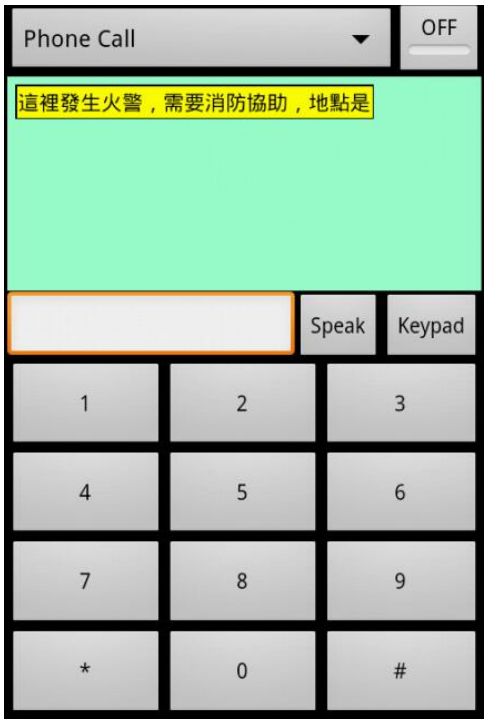
4.1.3 User Interface

The user interface of InCom looks very similar to those of other messaging applications. At the top of the screen, it consists of a voice source option list, which allows users to choose which voice source is recorded, phone call or microphone. An On/Off button that starts or stops the real-time speech-to-text feature is located to the right of the option list.

At the middle of the screen, there is an area that displays transcribed text and message sent by the user.

At the bottom of the screen, there is an input field, and a button that allows the user to send his/her message as voice. Furthermore, a button to show/hide a keypad with button 0 to 9, *, # for responding to telephone hotline IVR systems is located at the bottom right of the screen.





4.1.4 Dual-Tone Multi-Frequency Signaling (DTMF) Input

A keypad UI that allows user to input DTMF signal to IVR system has been made as mentioned in the previous paragraph. The necessary class in the Android SDK to generate DTMF signal has been identified. This feature is expected to be completed by the next phase.

4.1.5 Speech-to-Text

4.1.5.1 Speech Recognition Engine

The usability of the Cantonese recognition engine of the Microsoft Speech Platform remains untested. Grammars for a selected IVR system will be built and the engine will be tested by the first half of the next semester.

Windows 7, but not Windows Server 2008 R2, has a Putonghua (普通話) and a Guoyu (國語) speech recognition engine with dictation ability if the Chinese (PRC) and the Chinese (Taiwan) language packs are installed respectively. The engines will be tested

early next semester, and ways to run them on Windows Server 2008 R2 will be found out as soon as possible if any of these engines is to be used.

Search for other speech recognition engines continues.

4.1.5.2 Web Service

The speech-to-text web service will use HTTP. However, getting the result text with the recorded audio being streamed to the server at the same time may be a challenge, as a HTTP server responds only after the client finishes its request. InCom keeps sending any recorded audio to the server until the user stops it, and the server has to split up the recording into different piece for speech recognition and return the results back. If the user has to wait for whole recording uploaded, processed and returned by the server, he/she will feel that the application is very unresponsive.

There are 2 possible solutions to address the issue:

- Use 2 HTTP connections:
    1. Client posts a request to the web service entry point.
    2. The web service entry point creates a session for the client, and returns 201 Created response with URL to the session.
    3. Client keeps posting recording to the session URL using chunked transfer encoding, and then makes a get request to the same URL.
    4. Server returns the results using chunked transfer encoding.
- Use WebSocket (RFC 6455)

WebSocket is a relatively new technology that supports bi-directional traffic with 1 single TCP connection. Since it uses fewer network resources, it is currently preferred over the 2-HTTP-connections solution. It is not supported by the current version of Windows and .NET Framework. Some libraries that implement WebSocket are currently being evaluated. We may also implement WebSocket ourselves if necessary.

4.1.6 Network Data Usage

Internet connection is not free and is usually limited for mobile network. The lossless WAVE format is known to be large and expected to consume a lot network data usage. To reduce the size of data transfer, DEFLATE compression (RFC 1951) is used in web services. More information on data usage of InCom will be collected in the next phase.

4.1.7 Known Issue

4.1.7.1 LPCM Data Byte Order

The 16-bit LPCM data in the WAVE format uses little-endian while L16 defined in RFC 3551 uses network byte order (big-endian). Documentation of the `AudioRecord` class does not specify which byte order is returned. It is expected that the byte order used follows the endian of the hardware. The phone we bought is little-endian. However, some phone models, according to online sources, may be big-endian. As we do not have a big-endian phone for development and testing, this issue cannot be confirmed.

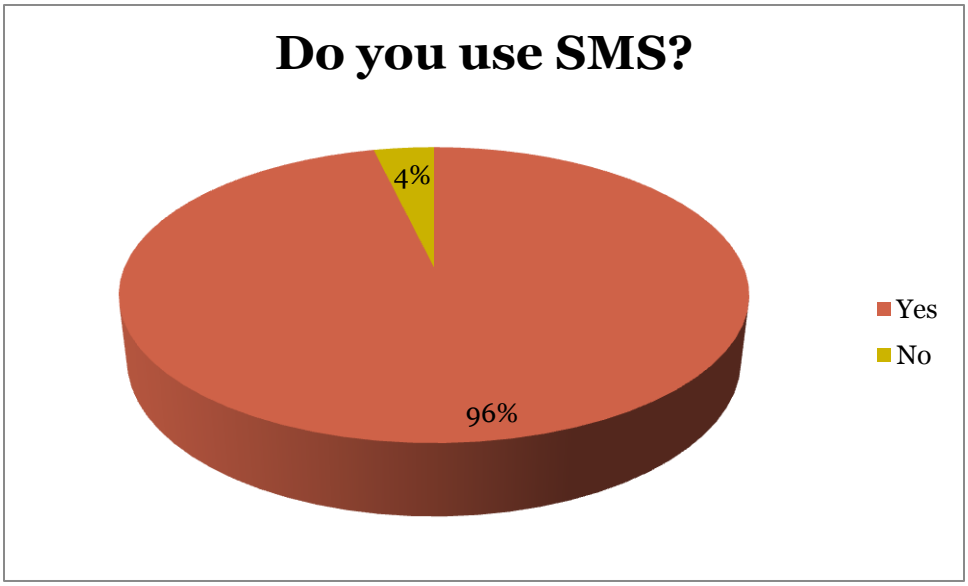
4.2 SS

SS stands for Smart Sentence which is an input method embedded in InCom. Once users have installed InCom on their phone, they can use SS in other applications such as Short Message System (SMS) and WhatsApp. SS is a user-friendly and efficient input method because users can construct a grammatically correct sentence easily by choosing corresponding words and phrases. There is no need for users to learn a set of rules for inputting words and phrases.

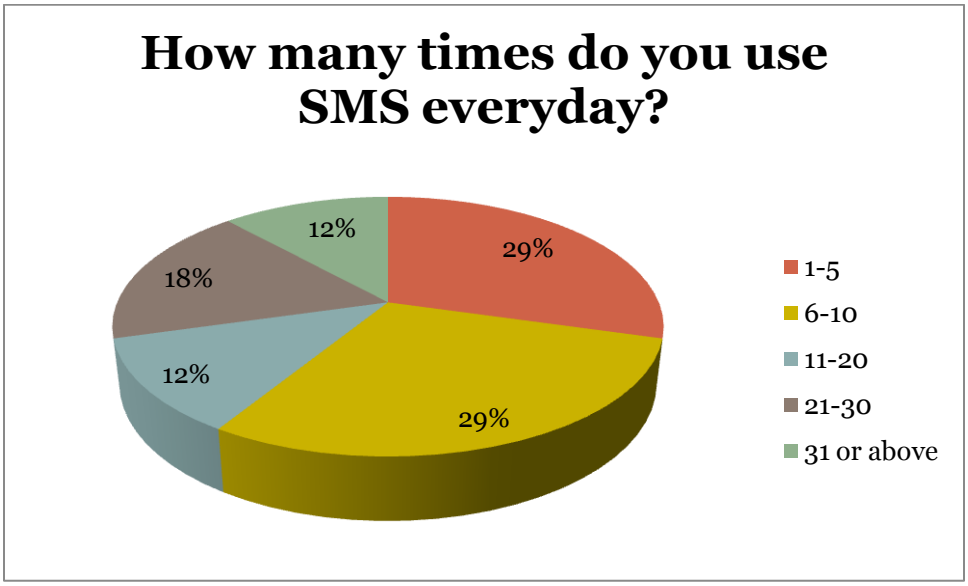
4.2.1 Sentence Structure Research

In order to investigate the way to categorize Chinese words or phrases, we have conducted a sentence structure research in November 2011. Questionnaires entitled "Survey on the habit of using SMS" were distributed to the hearing impaired (sample questionnaire is attached in appendix I). We sent 30 questionnaires to Hong Kong Association of the Deaf, in the meantime, published online version via http://tinyurl.com/ckm6syn. Finally, 27 responses were received. Here are the findings from the research:

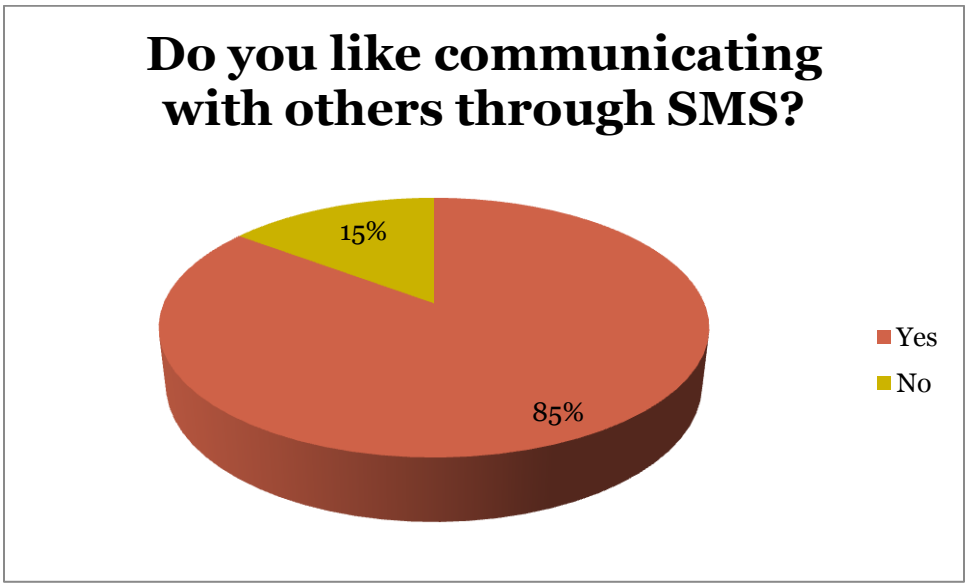
Almost all the hearing impaired uses SMS.



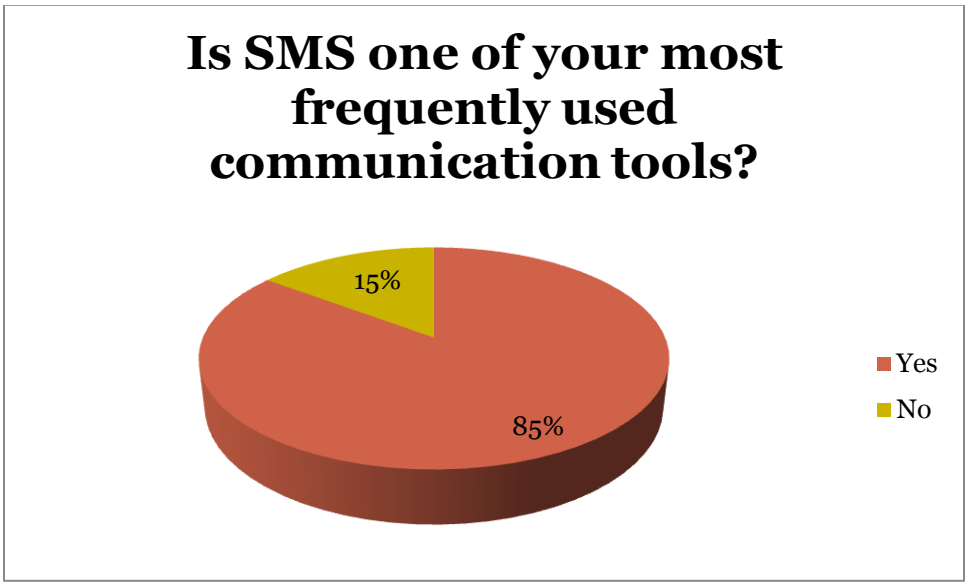
The hearing impaired use SMS several times a day. More than half of them use SMS 1 – 10 times a day. Some of them even use it more than 31 times a day.



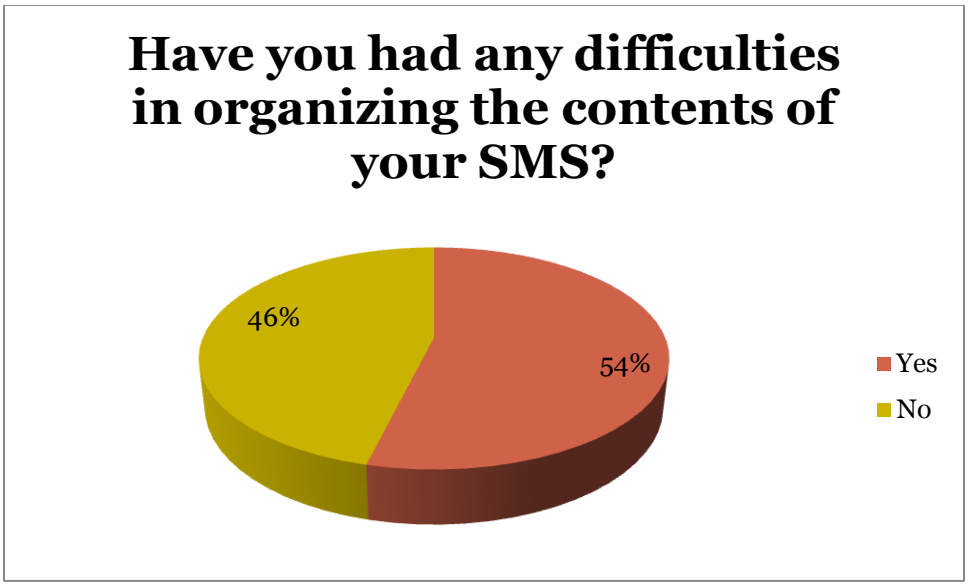
Most of the hearing impaired enjoy communicating with others using SMS.



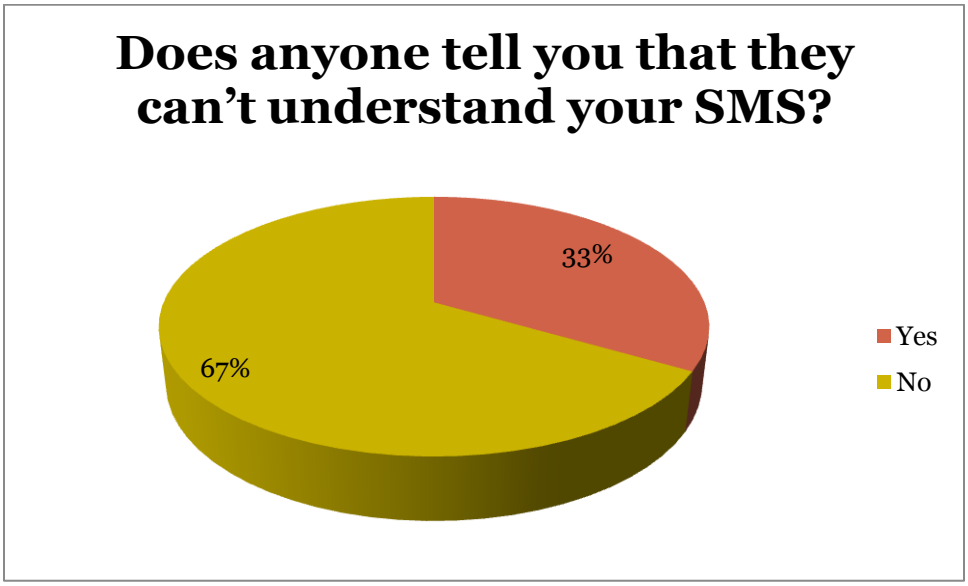
Around 85% hearing impaired rate SMS as one of their most frequently used communication tools. As they enjoy and rely heavily on SMS to communicate with others, we hope they can use our input method – SS to compose their SMS. In this way, their communication abilities can be enhanced.



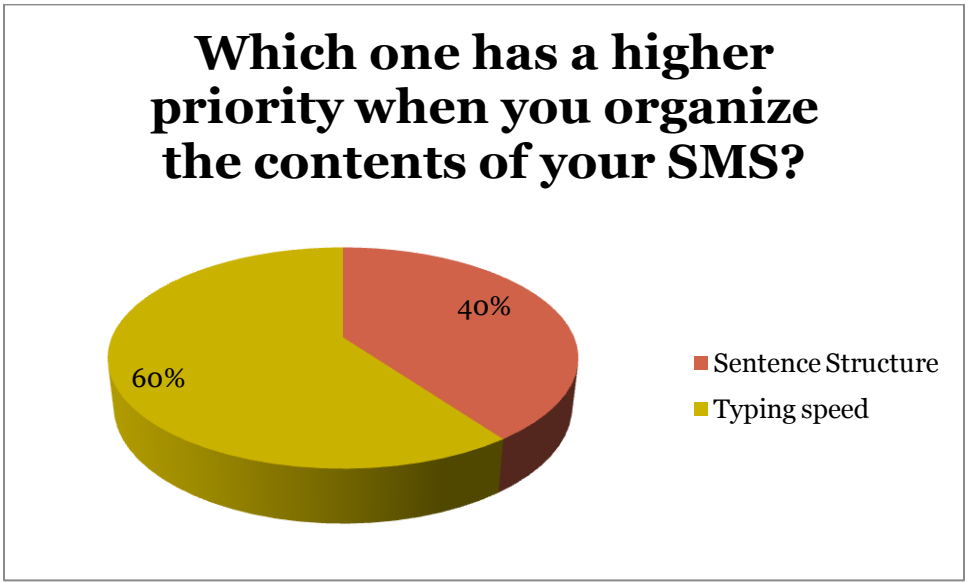
Also, in our research, we found that a lot of the hearing impaired have had difficulties in organizing the contents of their SMS because they have difficulty in expressing their feelings in the text, they are lack of vocabularies or they have difficulty in inputting text.



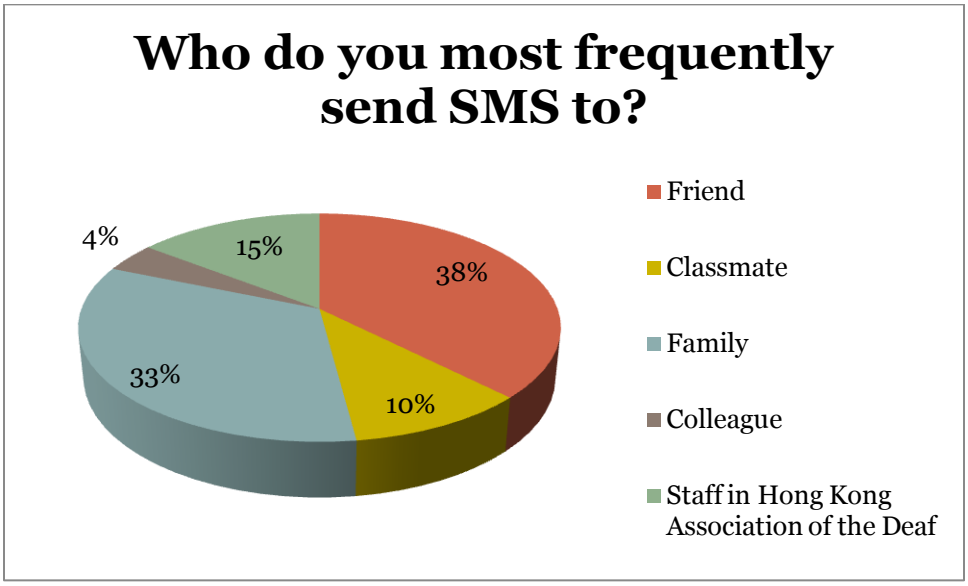
About 33% of the hearing impaired have been told their SMS contents are not understandable because of incorrect sentence structure or incorrect grammar.



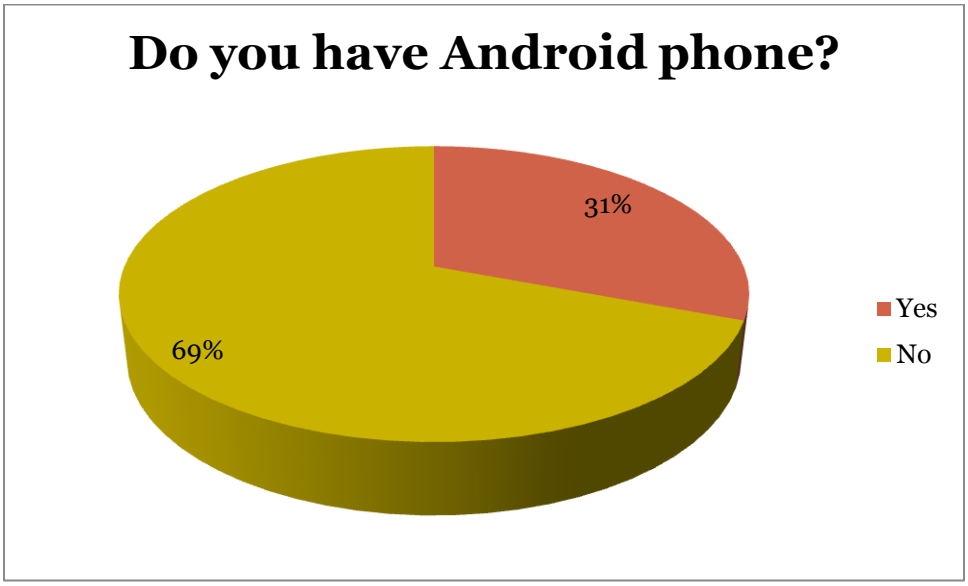
We found that the hearing impaired rather concern typing speed than sentence structure when they organize their SMS contents. This is another reason that their SMS contents are not understandable by others. Therefore, SS aims to tackle these problems by providing word and phrase buttons for them to choose.



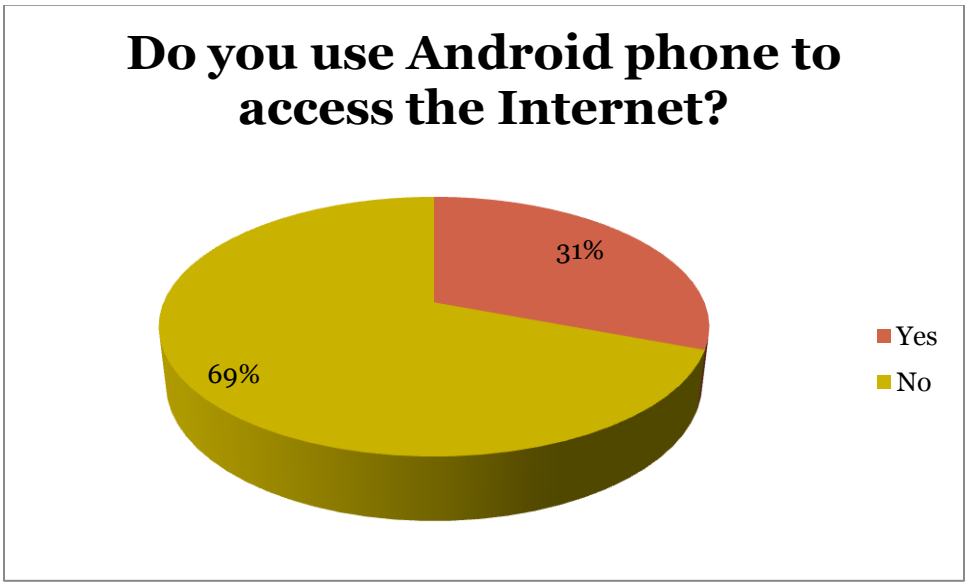
In the following question, we found that the hearing impaired usually send SMS to their friends and family. And the contents of their SMS are mainly about dating, invitation, greeting their friends to show supports and caring, travelling, chatting and reply. We are going to categorize our word and phrase buttons of SS base on this finding. Implementation of these categories into SS is expected to be done in the next semester.



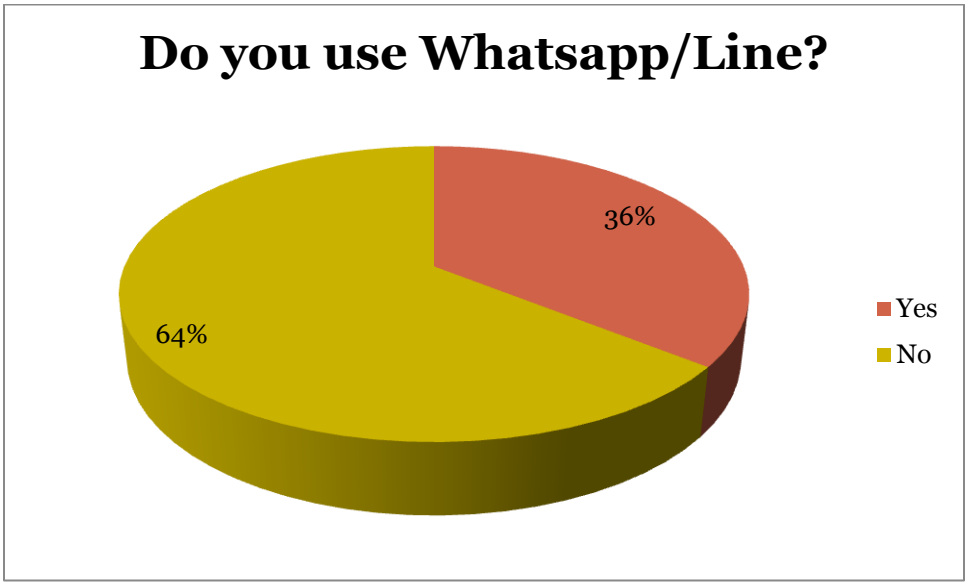
In the last few questions of the questionnaire, we found that only 1/3 of the hearing impaired have Android phone. However, we believe that Android is getting more popular in Hong Kong as the price of Android phones is constantly dropping.



Among those have Android phone, they all use their Android phone to access the Internet. The result is not surprising because the use of mobile Internet connection and wifi are popular in Hong Kong. Thus, the necessity of Internet connection of InCom should not be a problem for those who have Android phone.



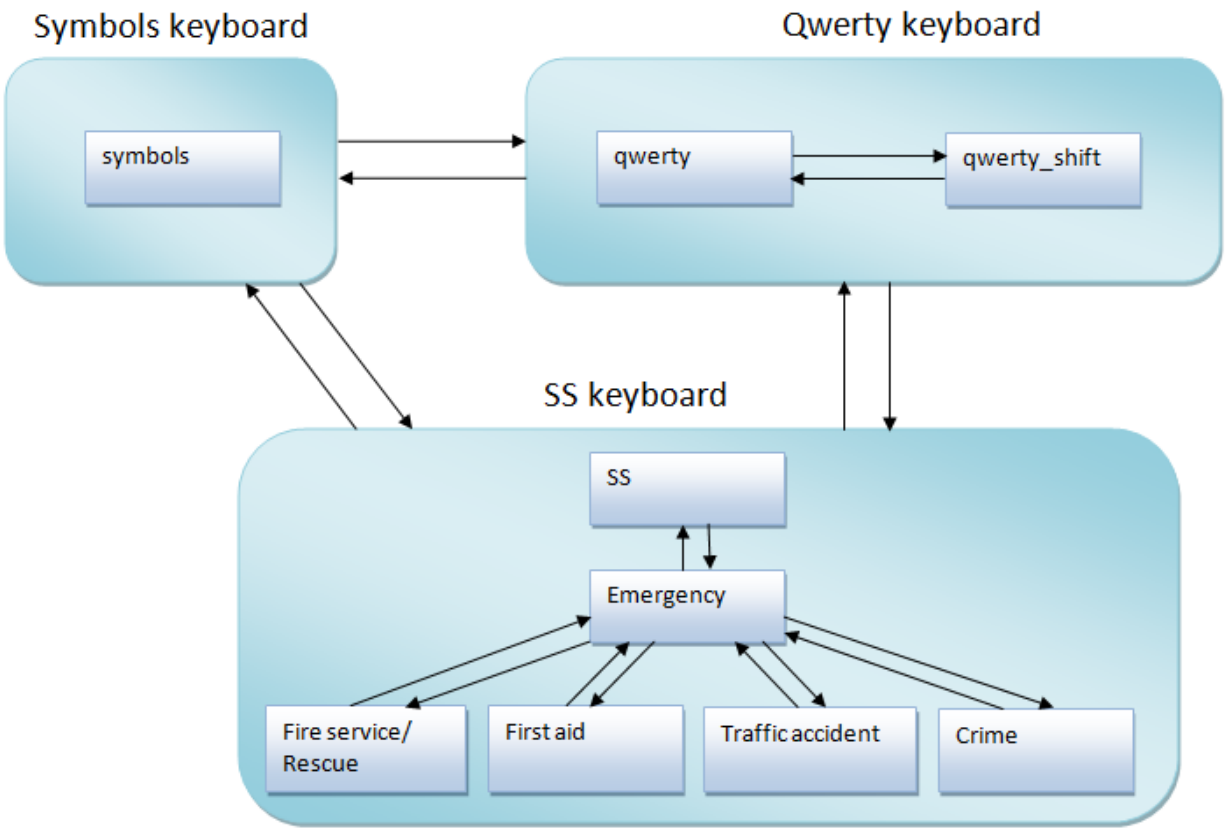
In the last question, we found that around 1/3 of the hearing impaired use Whatsapp/Line. The results show that messaging applications are quite popular among the hearing impaired. As the user interface of InCom looks very similar to these messaging applications, we hope that it can help our users more familiar with InCom.



4.2.2 Emergency Service

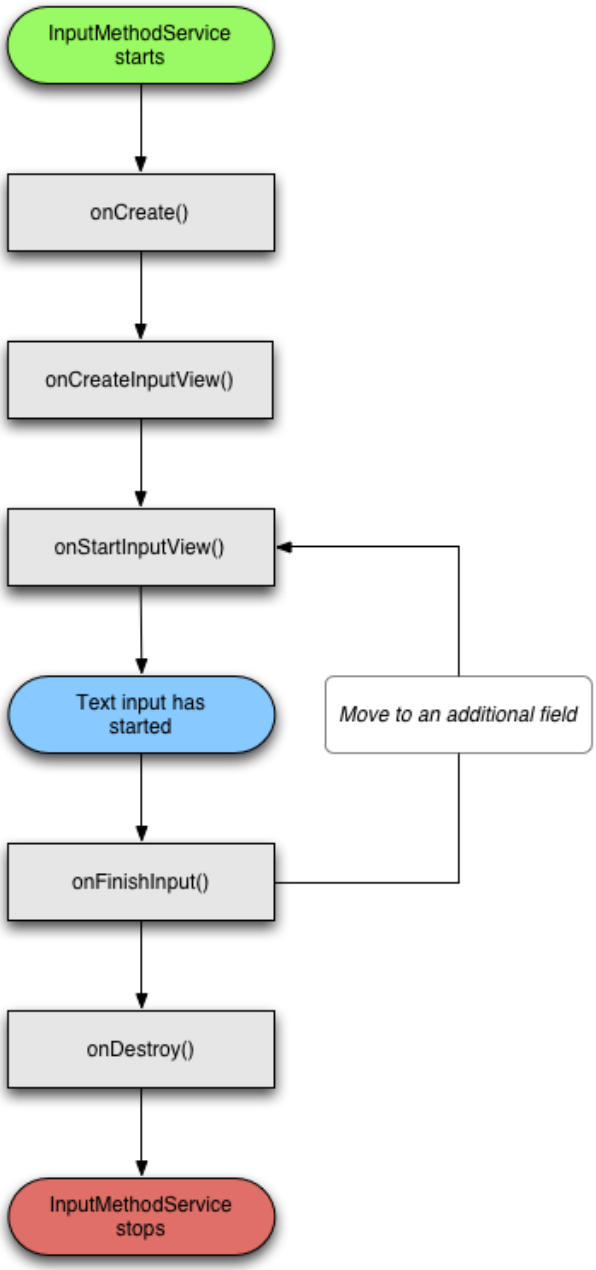
One of the word/phrase buttons is “Emergency” button. This is a special button explicitly for constructing message for 992 Emergency SMS services. When user presses the “Emergency” button, types of assistance user needs will be shown. After choosing the type of assistance needed, the natures of incident will be shown for user to choose. The corresponding text will then be automatically generated in the text area. In the meantime, the detailed address of the incident will be shown by tracing the location of the phone.

4.2.2.1 System Architecture



4.2.2.2 System Implementation

In order to create an input method on Android phone, we need to extend `android.inputmethodservice.InputMethodService` and implement `android.inputmethodservice.KeyboardView.OnKeyboardActionListener`. `InputMethodService` provides much of the basic implementation for SS. Here is the system flow of an `InputMethodService`:



When the input method service - SS is first created, `onCreate()` is called to initialize the service. When SS is displayed for the first time, `onCreateInputView()` is called. Input view is where user can input text either in form of key presses, handwriting or other gestures. `onCreateInputView()` creates and returns the view hierarchy to be displayed for SS, that is to initialize all the keyboard views in SS.

Different text fields will have different input types specified on them, such as number, date, phone and text. `onStartInputView()` returns different input view due to different nature of the text fields. If the input types are number, date or phone, symbols keyboard will be shown. For all other input types, SS keyboard will be shown by default.

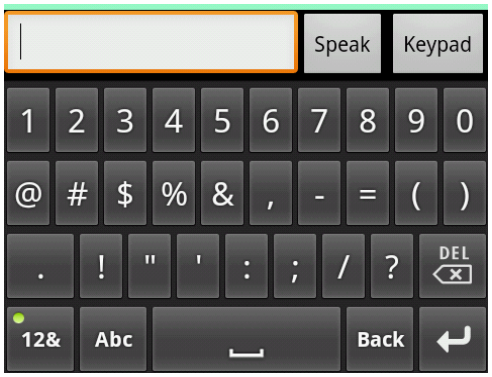
When the text input has started, text is sent to the application's text field via handling individual key events. Different keys will trigger different key events, thus, different responses will be shown.

When the text input has finished in the last editor, `onFinishInput()` is called. At this point there may be a call to `onStartInput()` to perform input in a new editor, or SS may be left idle. However, `onFinishInput()` is not called when input restarts in the same editor. When SS is no longer used and is being removed, `onDestroy()` is called. Also, all the resources SS holds will be cleaned at this point. Upon return, there will be no more calls in to SS and it is effectively dead.

4.2.2.3 User Interface

The user interface of SS looks very much alike other Android soft keyboard. It is implemented in Java xml file. As shown in system architecture, SS is mainly composed of three keyboards, namely symbols keyboard, qwerty keyboard and SS keyboard. Users can interchange these three keyboards by pressing the corresponding change mode buttons (i.e. "12&", "Abc" and "SS") in the lower left hand side of the keyboard.

Below is the symbols keyboard, it is responsible for numbers and symbols:



Here is the qwerty keyboard, it shows lowercase English letters. When users press the shift button, uppercase English letters will be shown:

Below is the SS keyboard. When users press the emergency button, four types of assistance will be shown:

4.2.2.4 Categories in Emergency Service

There are several categories in Emergency Service. These categories are divided based on the 992 Emergency SMS User Guide. The following parts will explain them one by one.

4.2.2.4.1 Fire Service / Rescue

This category requests service from the Fire Services Department. It is further divided into few sub-categories, such as Gas Leakage, Drown, Trapped, Fire and Lost. Here is the architecture of this category.

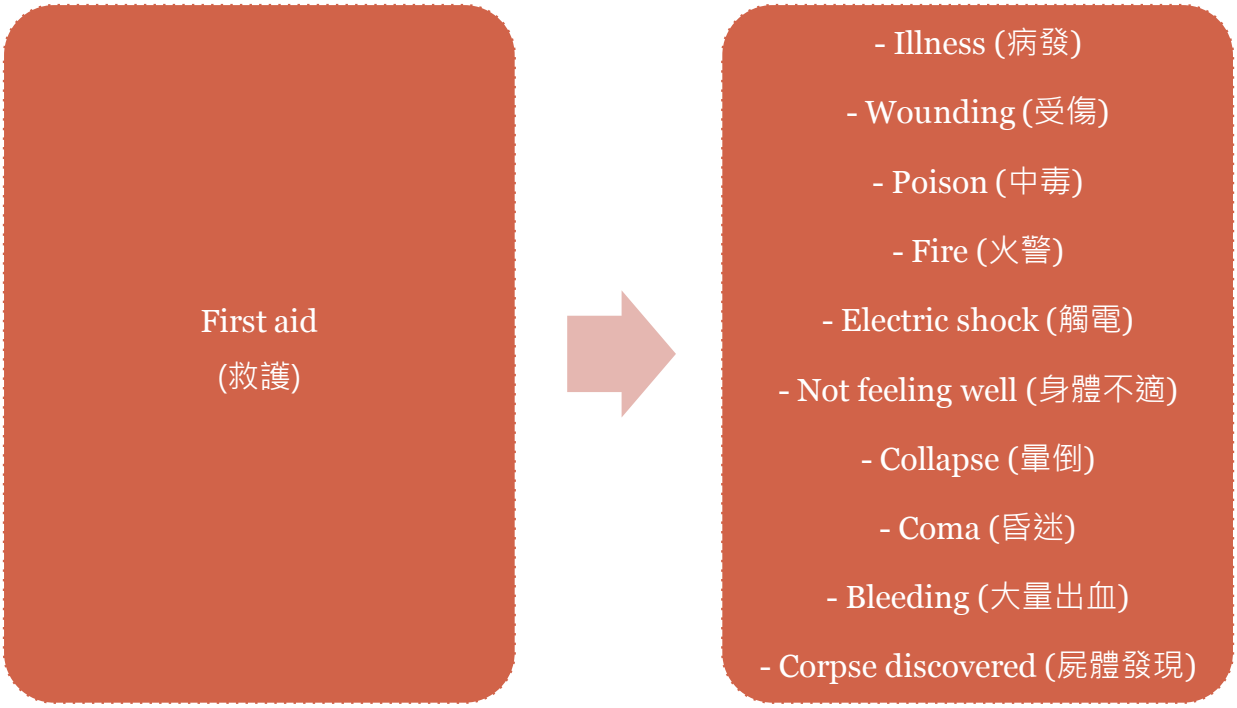
The sentence will be constructed when the user select one of these sub-categories. Below are the screenshot of it and the sample sentence that is constructed.

Lost(迷路)

這裡有人迷路，需要消防協助，地點是香港西環薄扶林道 79 號

4.2.2.4.2 First Aid

This category requests ambulance service. There are several types of injuries for user to select. It is different from the Fire Service / Rescue category because in an accident, several types of injuries can be happened at the same time. Therefore, this category allows user to multiple select the types of injuries. Below is the architecture of this category.



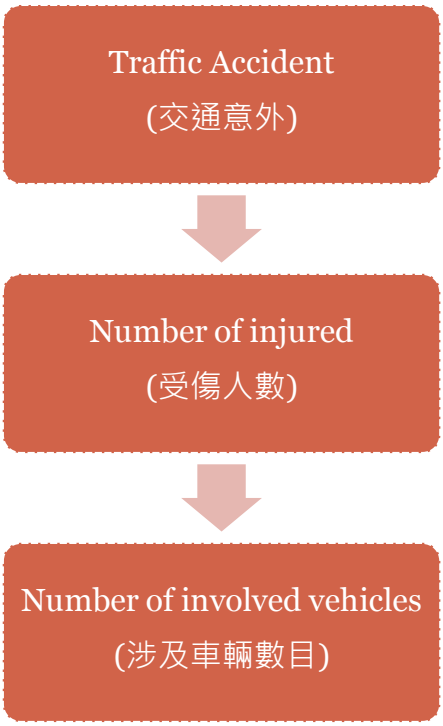
The sentence will be constructed when the user press the Next button or the Corpse discovered (屍體發現) is selected. The screenshot and the sample constructed sentence are shown below.

Poison (中毒), Coma (昏迷)

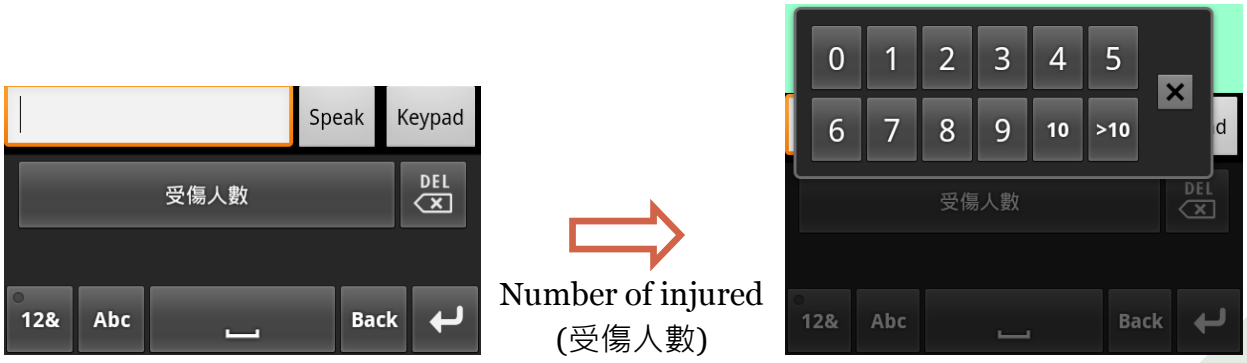
這裡有人中毒，昏迷，需要救護車入醫院，地點是香港西環薄扶林道 79 號

4.2.2.4.3 Traffic Accident

This category requests service mainly from the Police Force. In this category there are two more sub-pages that allow the user to select the number of injured and the number of involved vehicles. The architecture of this category is as follow.

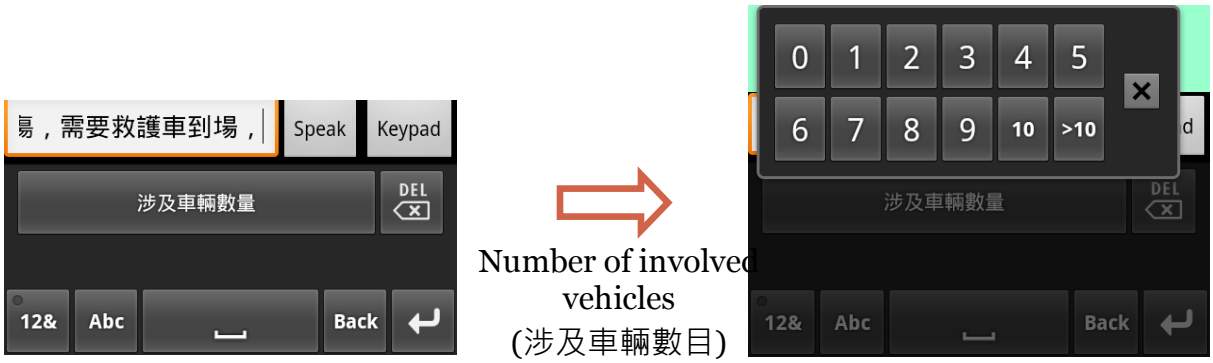


When the buttons Number of injured and Number of involved vehicles are being selected, there will be a pop up keyboard that lets the user to select the corresponding number. This pop up keyboard is done by adding an attribute called "android:popupKeyboard" in the Key element of keyboard xml file, and point it to the corresponding pop up keyboard xml file. The following are the screenshot of this category and the constructed sentence.



After the Number of injured has been selected, the first part of the sentence will be shown. For instance, if there are 3 people being injured, the first part of the sentence will be: “這裡發生交通意外，有 3 人受傷，需要救護車到場，”

Afterward, select the Number of involved vehicles and finish constructing the sentence.

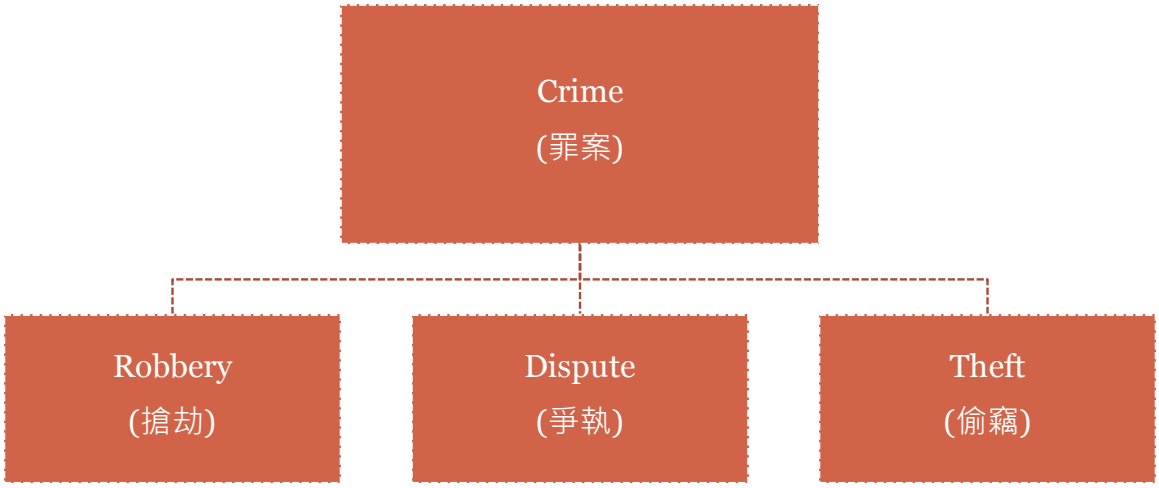


Number of involved vehicles
(涉及車輛數目)

If there are 2 vehicles involved in the accident, the sentence will be constructed as: “這裡發生交通意外，有 3 人受傷，需要救護車到場，這意外涉及 2 架車輛，地點是香港西環薄扶林道 79 號”

4.2.2.4.4 Crime

This category requests service from the Police Force. It is similar to the Fire Service / Rescue category. It is further divided into a few sub-categories, namely Robbery, Dispute and Theft. Here is the architecture of this category.

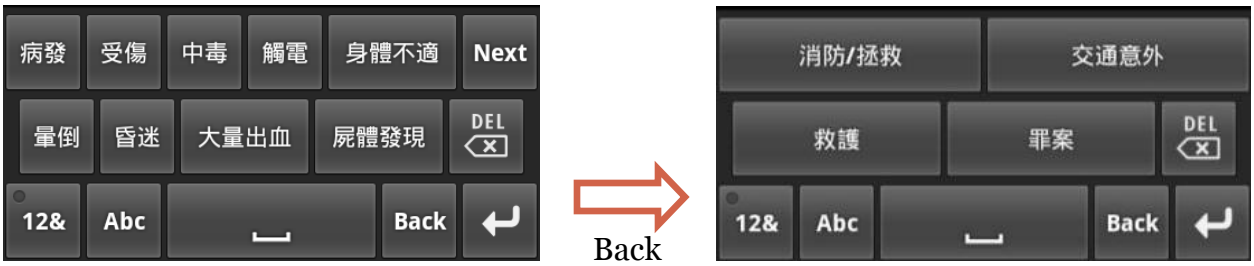


The sentence will be constructed when the user select one of the sub-categories. The screenshot and the sample constructed sentence are shown below.

Robbery (搶劫)　這裡有人搶劫，需要警察到場，地點是香港西環薄扶林道 79 號

4.2.2.5 Keeping the Keyboard State

Since there will be a lot of pages and sub-categories within SS, from time to time, user may want to move back to the previous page if they could not find the category that they want or they simply press the wrong button. In SS, there is always a Back button located at the bottom of the keyboard. The screenshot below is one of the illustrations.

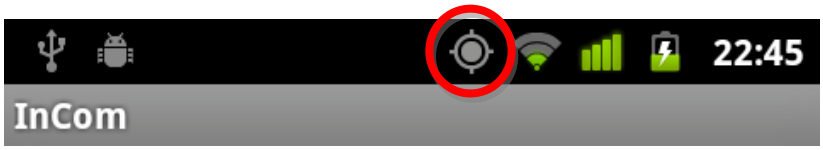


In order to keep the keyboard state, the class Deque is used when we are implementing it. The keyboard state will be pushed when we move to the next page or category, while it will be popped when we go back to the previous page. Since the keyboard can always be changed to the symbols keyboard and the qwerty keyboard, it was one of the difficulties when implementing it.

4.2.2.6 User Location

For the last part of the constructed sentence in Emergency Service of SS, it always shows the address of the user. Indeed, this is done by tracing the user location when they are constructing the sentence. The aim of doing it is to speed up the input speed of the user. In Android, it is mainly done by two classes, which are `android.location.LocationManager` and `android.location.Geocoder`. The `android.location.LocationManager` class is responsible for collecting the user location information, such as latitude and longitude. There are two kinds of sources provided in Android SDK to collect the user location information, GPS and NETWORK_PROVIDER. GPS determines location using satellite, while NETWORK_PROVIDER determines location based on availability of cell tower and Wi-Fi access points. The `android.location.Geocoder` class is used to send the latitude and longitude information to Google and convert them into address. The address, however, is no way to make sure that it is one hundred percent accurate. Therefore,

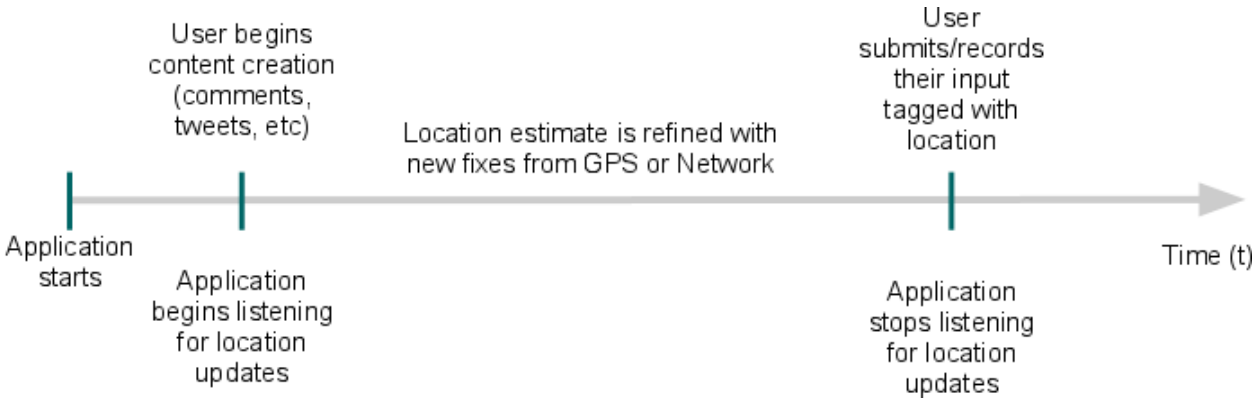
manual adjustment is required. When the device is collecting the user location using GPS, the following figure will be shown on the notification bar.



When there is no internet connection or location information, there will be a small pop up message to inform the user. The user should then check if they have switched off the Wi-Fi or GPS.

4.2.2.6.1 Concerns about tracing the location

There are some concerns when implementing this part. One of them is the trade-offs in accuracy, speed and battery-efficiency. Although tracing the user location for a longer time means that the address will be more accurate, it consumes more battery. Therefore we adopt a model as follow:


User begins content creation (comments, tweets, etc)
Location estimate is refined with new fixes from GPS or Network
User submits/records their input tagged with location
Application starts
Application begins listening for location updates
Application stops listening for location updates
Time (t)

When SS is started, it will not start tracing the user location. It is only started when the user select the Emergency Service category. It stops after the users constructed the sentence or they move back to the first page of SS.

Another concern is the variance in accuracy. Since latest location information does not necessary means that it is more accurate, it has to determine which location information should be used based on their accuracy and information collected time. In the implementation, there is a method called `isBetterLocation`, which is responsible for checking which location information should be used.

5 Dependencies

Both speech to text and text to speech conversion in InCom depend on Microsoft Speech Platform for Servers that is hosted on a CS virtual server running Windows Server 2008 R2.

6 Software Documentation

6.1 Application Manual

A manual will be implemented inside the application to teach users how to use our application.

6.2 Javadoc

Javadoc will be prepared to provide better maintenance.

7 Quality Assurance

We want our application to fit the needs of the hearing impaired and work properly across different brands of Android phones. We plan to take the following measures to ensure the quality of our application:

7.1 Unit Testing

Test cases will be created as appropriate. However, as our application has non-textual content, it is difficult to use an automated unit testing framework to conduct testing automatically. We will test our application manually.

7.2 Revision Control

The source code of the project has been hosted using Visual SVN, an Apache Subversion (SVN) server. Eclipse has better support to SVN compared to Microsoft Team Foundation Service, which is currently in preview state. With a revision control system, the changes of the source code can easily be tracked and reverted in case of any regression. It also helps to avoid unexpected overwriting when more than one team mates are working on the same part.

7.3 Usability Testing

Preview versions will be released to members of the Hong Kong Association of the Deaf to collect user feedback.

8 Financial Report

The table below summarizes the financial report of our project up to 9 Jan 2012:

		Unit Price	Quantity	Amount	Total
Income					
Budget from the Department of Computer Science		$1000	4	$4000	
					$4000
Less:	**Expenditure**				
	Android Phone	$1998	1	$1998	
	Windows Server 2008 R2	$700	1	$700	
					$2698
					$1302

Initially, each student taking CSIS0801 Final Year Project is granted $1000 of budget. As we have four members in our team, we have a total budget of $4000 for buying equipment for our project.

Our team is developing an Android application which requires Android version 2.3 to support the speech recognition function. Moreover, the emulator provided in the Android SDK does not support audio recording and speech recognition. Therefore we have purchased a new Android phone – Sony Ericsson Xperia mini pro – for development.

In addition, as many Android phones do not support Cantonese text-to-speech by default, a Windows Server 2008 R2 was purchased from the Department of Computer Science to host a web service using Microsoft Speech Platform for Servers to perform text-to-speech conversion. The server will also be used to host speech-to-text conversion service in the future.

9 Schedule

Phase 0
- Scope Definition
- Background Research
- Project Plan and Project Webpage
- (By 23 Sep 2011)

Phase 1
- Development Environment Setup
- Feasibility Study
- Sentence Structure Research (continuous)
- (By 23 Oct 2011)

Phase 2
- Graphical User Interface
- Sentence Structure Research
- Smart Sentence (Emergency Service)
- InCom (Text-to-Speech)
- (By 31 Dec 2011)

Phase 3
- First Presentation
- Interim Report
- (By 9 Jan 2012)

Phase 4
- Smart Sentence
- InCom (Speech-to-Text)
- System Testing
- Documentation
- (By 31 Mar 2012)

Phase 5
- Final Report
- Final Presentation
- Project Exhibition
- (By 23 Apr 2012)

10 Work Division

The following is a list of our project team members and our responsibilities in this project:

Name	Responsibilities
Lau Kam Lun	• Feasibility Study • Smart Sentence
Law Chi Lok	• Sentence Structure Research • Graphical User Interface • InCom
Tse Shing Chi	• Feasibility Study • InCom
Wong Wai Ling	• Sentence Structure Research • Smart Sentence

Below is a list of responsibilities performed by all of our team members:

All members	• Development Environment Setup • System Testing • Documentation

11 Coming Development

In the coming semester, there are several important features that we have to implement. The first one is the speech to text part, which is a major component in InCom. In addition, the user interface has to be polished, since the interface at the moment is just at the very beginning stage. We would like to make it more attractive, like other instant messaging applications.

Another feature that is going to be implemented is the categories in SS. Currently, we only have the “Emergency” button. In the next semester, we would like to implement a few more categories for it. According to the sentence structure research, categories like dating, greeting and inviting will be implemented.

Finally, a lot of testing has to be done in order to make sure that application functions properly. Moreover, we will create a user manual to let user know how to use InCom and provide documentation.

12 Conclusion

Up till now we have started our project for a semester. As this project is our first time to develop an android application, before we start to develop our project, we have preformed feasibility study to study the knowledge on writing an application with voice processing and contacted the Hong Kong Association of the Deaf to study for their needs and performed a sentence structure research on the hearing impaired.

For the implementation on the application InCom, we have developed the basic version of the user interface which included the buttons needed and a display area. In addition, the text to speech part of the InCom has been completed.

We have implemented the Emergency Service of the SS keyboard which includes 4 types of assistance – Fire Service/Rescue, First Aid, Traffic Accident, and Crime. The user location can be traced by the GPS and network information.

For the coming semester, we will proceed with our development and hope that our application will enhance the communication ability of the hearing impaired.

13 References

1. Hong Kong Association of the Deaf
   www.hongkongdeaf.org.hk
2. 992 Emergency SMS User Guide
   http://www.police.gov.hk/ppp_en/02_er_room/doc/smsemg.pdf
3. Android Developer – Speech Input
   http://developer.android.com/resources/articles/speech-input.html
4. Microsoft Research – Project Haiwaii
   http://research.microsoft.com/en-us/um/redmond/projects/hawaii/students/default.aspx

14 Appendix I: Questionnaire

問卷調查 – 有關手提電話短訊(SMS)的使用習慣

你好，我們是香港大學計算機科學系的同學，現正進行有關手提電話短訊(SMS)使用習慣的調查，希望你可以幫忙完成這份問卷調查，謝謝。

Q1. 你有使有短訊的習慣嗎?
   1. 有
   2. 沒有

Q2. 你使用短訊的頻率?
   _____次/日

Q3. 你喜歡用短訊和別人溝通嗎?
   1. 喜歡
   2. 不喜歡

Q4. 短訊是否你常用的溝通工具?
   1. 是
   2. 不是

Q5. 你最常發送短訊的對象類別依次為: (1(最常) – 5(最少))
   朋友_____ 家人_____ 中心職員_____
   同學_____ 同事_____ 其他(請註明):_____

Q6. 你發送短訊的內容主要為: (例如: 邀約朋友聚會)
   1. _____
   2. _____
   3. _____
   4. _____

Q7. 你在組織短訊內容時有否遇過困難?
   1. 有，原因:_____
   2. 沒有

Q8. 別人有否表示過看不懂你發出的短訊?
   1. 有，原因:_____
   2. 沒有

Q9. 你在組織短訊時會比較注重句子結構還是打字效率?
   1. 句子結構

2. 打字效率

Q10. 你有使用 Android 電話嗎?

1. 有
2. 沒有

Q11. 你有使用 Android 電話上網嗎?

1. 有
2. 沒有

Q12. 你有使用 Whatsapp / Line 的習慣嗎?

1. 有
2. 沒有

多謝你完成這份問卷！如有任何查詢，歡迎電郵至 fyp11001@cs.hku.hk。